fhicl-cpp - Bug #23656

Parsing of value silently fails: oops

11/26/2019 01:08 PM - Gianluca Petrillo

Status:	Closed	**Start date:**	11/26/2019
Priority:	Normal	**Due date:**	
Assignee:	Kyle Knoepfel	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	3.06.00	**Spent time:**	2.00 hours

Description

Attempting to write a regex into a FHiCL parameter, I found that the FHiCL parser "silently" replaces the value which failed parsing with "oops".
For example, a FHiCL file with content:

```
pattern: "$\d+^"
```

is seen by fhicl-expand like:

```
Policy is 1; path is "FHICL_FILE_PATH"
pattern: "$\d+^"
```

but fhicl-dump shows it as

```
# Produced from 'fhicl-dump' using:
#   Input  : pattern.fcl
#   Policy : cet::filepath_maker
#   Path   : "FHICL_FILE_PATH"

pattern: oops
```

I would rather see the parser throw an exception in these cases.

(I am sure there is a way to correctly escape these strings... I ended up being more standard and using [[:digit:]] instead of \d, which is just as good.)

Associated revisions

Revision 541160c5 - 05/14/2020 01:14 PM - Kyle Knoepfel

Resolve issue #23656: Do not silently fail on malformed string.

History

#1 - 12/02/2019 10:26 AM - Kyle Knoepfel

- Status changed from New to Feedback

Seems to be a bug. Have you tried this with an actual *art* job, and you still get the "oops" problem?

#2 - 12/02/2019 12:18 PM - Gianluca Petrillo

Yes. That was, in fact, the context in which I encountered the issue.

#3 - 05/13/2020 03:57 PM - Kyle Knoepfel

- Target version set to 3.06.00

- Assignee set to Kyle Knoepfel

- Status changed from Feedback to Assigned

#4 - 05/14/2020 01:25 PM - Kyle Knoepfel

- % Done changed from 0 to 100

- Status changed from Assigned to Resolved

This bug has been fixed with commit fhicl-cpp:541160c. Using your sample configuration, the exception thrown now looks like:

```
$ fhicl-dump test.fcl
---- Parse error BEGIN
  The string "$\d+^" is not representable as a canonical string.
  It is likely you have an unescaped (or incorrectly escaped) character.
---- Parse error END
```

And indeed, if you were to use the single-quoted string ('$\d+^'), or the double-escaped 'd' within the double-quoted string ("$\\d+^"), you would get what you want. Incidentally, assuming this represents a regular expression, the ^ and $ characters should probably be reversed.

#5 - 05/14/2020 02:19 PM - Gianluca Petrillo

Kyle Knoepfel wrote:

> Incidentally, assuming this represents a regular expression, the ^ and $ characters should probably be reversed.

I consistently confuse ^ and $. Kind of a branding thing for me; for the rest of the people, though... yes, they should really swap them.

#6 - 07/10/2020 01:58 PM - Kyle Knoepfel

- Status changed from Resolved to Closed